

## Θεωρία

### Διαδική Αναζήτηση

Ο αλγόριθμος της δυαδικής αναζήτησης (binary search) εφαρμόζεται μόνο σε πίνακες που έχουν ταξινομημένα στοιχεία. Αν τα στοιχεία δεν είναι ταξινομημένα τότε δεν μπορεί να εφαρμοστεί.

Ο αλγόριθμος λειτουργεί ως εξής:

- Βρίσκουμε το μεσαίο στοιχείο του ταξινομημένου πίνακα.
- Εάν το προς αναζήτηση στοιχείο είναι ίσο με το μεσαίο στοιχείο τότε σταματάμε την αναζήτηση αφού το στοιχείο βρέθηκε
- Εάν δεν βρέθηκε, τότε ελέγχουμε αν το στοιχείο που αναζητούμε είναι μικρότερο ή μεγαλύτερο από το μεσαίο στοιχείο του πίνακα. Αν είναι μικρότερο, περιορίζουμε την αναζήτηση στο πρώτο μισό του πίνακα (με την προϋπόθεση ότι τα στοιχεία είναι διατεταγμένα κατά αύξουσα σειρά), ενώ αν είναι μεγαλύτερο περιορίζουμε την αναζήτηση στο δεύτερο μισό του πίνακα.
- Η διαδικασία αυτή λοιπόν επαναλαμβάνεται για το κατάλληλο πρώτο ή δεύτερο μισό πίνακα, μετά για το 1/4 του πίνακα κ.ο.κ. μέχρι, είτε να βρεθεί το στοιχείο, είτε να μην είναι δυνατό να χωρισθεί ο πίνακας περαιτέρω σε δύο νέα μέρη.

**ΠΡΟΓΡΑΜΜΑ** δυαδική\_αναζήτηση

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** A[20], Left, Right, M, k, S, i

**ΛΟΓΙΚΕΣ:** f

**ΑΡΧΗ**

**ΓΡΑΨΕ** 'Οι αριθμοί που θα δοθούν πρέπει να είναι ταξινομημένοι κατά αύξουσα τάξη'

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 20

**ΓΡΑΨΕ** 'Δώστε το', i, ' στοιχείο του πίνακα'

**ΔΙΑΒΑΣΕ** A[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Δώστε την τιμή που αναζητάτε '

**ΔΙΑΒΑΣΕ** S

Left <- 1

Right <- 20

k <- 0

f <- **ΨΕΥΔΗΣ**

**ΟΣΟ** (Left <= Right) **ΚΑΙ** (f = **ΨΕΥΔΗΣ**) **ΕΠΑΝΑΛΑΒΕ**

M <- (Left + Right) **DIV** 2

**ΑΝ** A[M] = S **ΤΟΤΕ**

k <- M

f <- **ΑΛΗΘΗΣ**

**ΑΛΛΙΩΣ**

```
ΑΝ A[M] < S ΤΟΤΕ
    Left <- M + 1
ΑΛΛΙΩΣ
    Right <- M - 1
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ f = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ "Το στοιχείο ", S, " υπάρχει στη θέση.", M
ΑΛΛΙΩΣ
    ΓΡΑΨΕ "Το στοιχείο ", S, " δεν υπάρχει στον πίνακα"
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Η **δυναδική αναζήτηση** να υλοποιηθεί με πρόγραμμα, όπως παρακάτω σε ταξινομημένο πίνακα 20 θέσεων. Πέρα από το τμήμα δηλώσεων, το πρόγραμμα έχει ένα επιπλέον τμήμα για το "γέμισμα" του πίνακα με στοιχεία (υποθέτουμε ότι ο πίνακας γεμίζει με σωστά ταξινομημένα στοιχεία σε αύξουσα σειρά).

**ΠΡΟΓΡΑΜΜΑ** δυναδική\_αναζήτηση

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** A[20], Left, Right, M, k, S, i

**ΛΟΓΙΚΕΣ:** f

**ΑΡΧΗ**

**ΓΡΑΨΕ** 'Οι αριθμοί που θα δοθούν πρέπει να είναι ταξινομημένοι κατά αύξουσα τάξη'

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 20

**ΓΡΑΨΕ** 'Δώσε το', i, ' στοιχείο του πίνακα'

**ΔΙΑΒΑΣΕ** A[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Δωσε τιμή για αναζήτηση: '

**ΔΙΑΒΑΣΕ** S

Left <- 1

Right <- 20

k <- 0

f <- **ΨΕΥΔΗΣ**

**ΟΣΟ** (Left <= Right) **ΚΑΙ** (f = **ΨΕΥΔΗΣ**) **ΕΠΑΝΑΛΑΒΕ**

M <- (Left + Right) **DIV** 2

**ΑΝ** A[M] = S **ΤΟΤΕ**

k <- M

f <- **ΑΛΗΘΗΣ**

**ΑΛΛΙΩΣ**

**ΑΝ** A[M] < S **ΤΟΤΕ**

Left <- M + 1

**ΑΛΛΙΩΣ**

Right <- M - 1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** f = **ΑΛΗΘΗΣ** **ΤΟΤΕ**

**ΓΡΑΨΕ** "Το στοιχείο,", S, "υπάρχει στη θέση:", M

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** "Το στοιχείο,", S, " δεν υπάρχει στον πίνακα"

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** δυναδική\_αναζήτηση

**Παράδειγμα**

Δίνεται ο πίνακας

1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Αναζήτηση του στοιχείου 38 (υπάρχει στον πίνακα)

Βήμα 1	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 2	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 3	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 4	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47

Με κίτρινο σημειώνεται το στοιχείο του πίνακα που εξετάζεται (στο μέσον)

Με πράσινο σημειώνεται το τμήμα του πίνακα που απομένει για αναζήτηση

Με κόκκινο σημειώνεται το τμήμα του πίνακα που έχει αποκλειστεί

(Τα ίδια χρώματα χρησιμοποιούνται και στο επόμενο παράδειγμα)

Αναζήτηση του στοιχείου 39 (δεν υπάρχει στον πίνακα)

Βήμα 1	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 2	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 3	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 4	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 5	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47